Assignment 0

Ahmed Yusuf

February 14, 2018

Github user name: ayusuf15

This Assignment is for us to learn and understand how to use Rstudio, Rmarkup and kintr along with GitHub. We used a tutorial by the title of “A (very) short introduction to R” by Paul Torfs & Claudia Brauer

Link to the document: https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf (https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf)

Setting working DIR to the one with all the scripts

```
setwd("/Users/ahmed/Documents/School/SNET4/SRT411/A0")
```

TODO Number 1

Compute the difference between 2014 and the year you started at this university and divide this by the difference between 2014 and the year you were born. Multiply this with 100 to get the percentage of your life you have spent at this university. Use brackets if you need them.

```
((2017-2014)/(2014-1993)) * 100
```

```
## [1] 14.28571
```

TODO Number 2

Repeat the previous ToDo, but with several steps in between. You can give the variables any name you want, but the name has to start with a letter.

```
year = 2014
birth = 1993
((2017-year)/(year-birth))*100
```

```
## [1] 14.28571
```

TODO Number 3

Compute the sum of 4, 5, 8 and 11 by _rst combining them into a vector and then using the function sum.

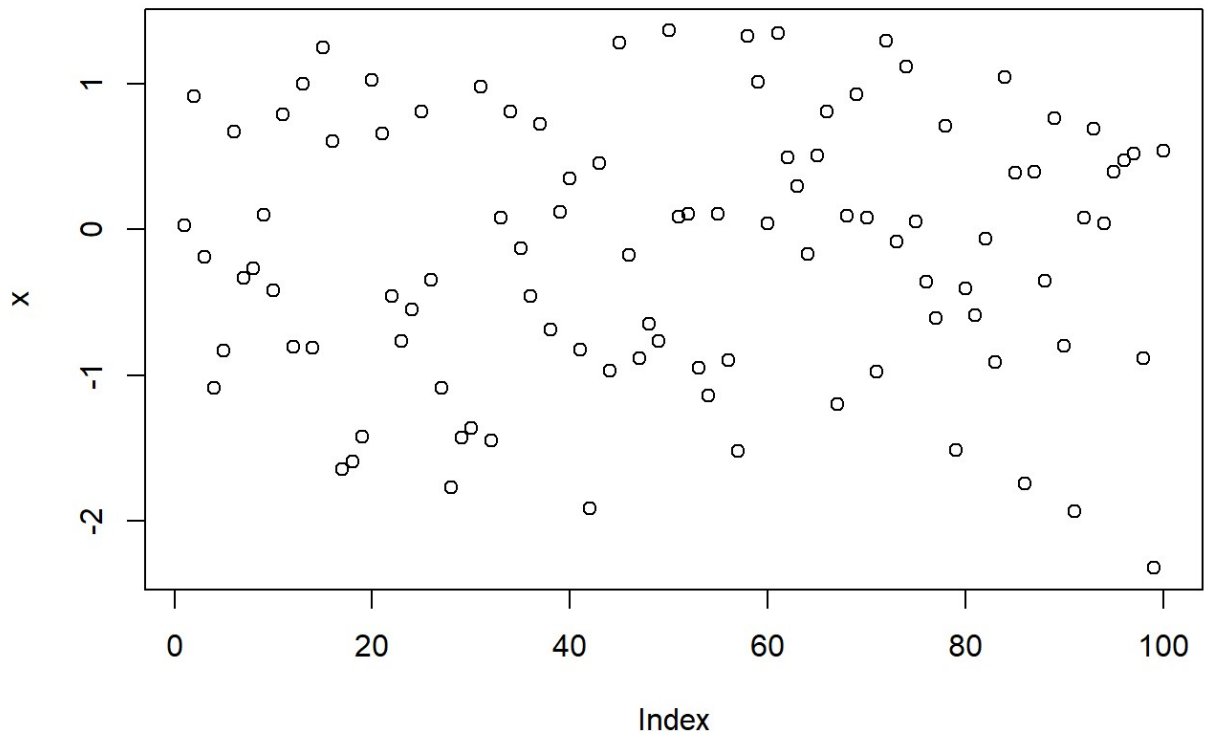
```
sum(c(4,5,8,11))
```

```
## [1] 28
```

TODO Number 4

Plot 100 normal random numbers.

```
x = rnorm(100)
plot(x)
```



TODO Number 5

Find help for the sqrt function.

```
help(sqrt)
```

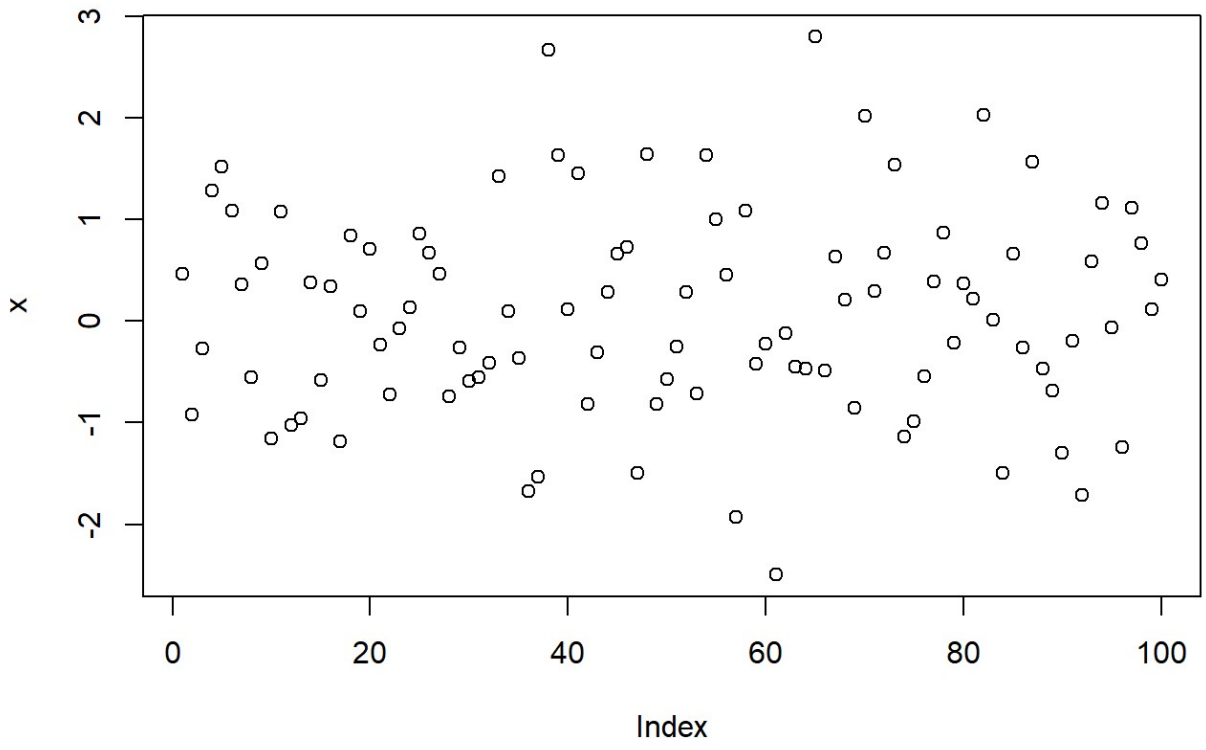
```
## starting httpd help server ... done
```

ToDO Number 6

Make a file called firstscript.R containing R-code that generates 100 random numbers and plots them, and run this script several times.

firstscipt.R x = rnorm(100) plot(x)

```
source("firstscript.R")
```



TODO Number 7

Put the numbers 31 to 60 in a vector named P and in a matrix with 6 rows and 5 columns named Q. Tip: use the function seq. Look at the difierent ways scalars, vectors and matrices are denoted in the workspace window.

```
p = seq(from=31, to=60)
q = matrix(p,ncol = 5, nrow = 6)
q
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   31   37   43   49   55
## [2,]   32   38   44   50   56
## [3,]   33   39   45   51   57
## [4,]   34   40   46   52   58
## [5,]   35   41   47   53   59
## [6,]   36   42   48   54   60
```

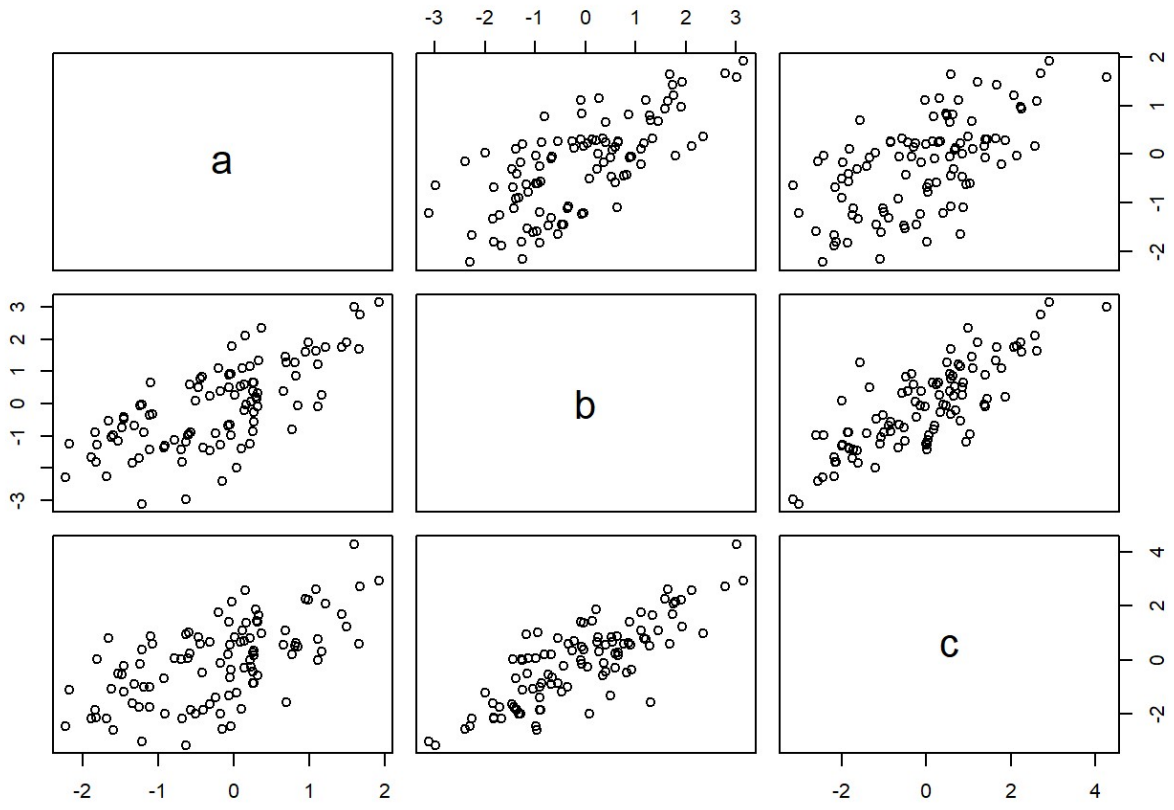
TODO Number 8

Make a script file which constructs three random normal vectors of length 100. Call these vectors x1, x2 and x3. Make a data frame called t with three columns (called a, b and c) con- taining respectively x1, x1+x2 and x1+x2+x3. Call the following functions for this data frame: plot(t) and sd(t). Can you understand the results? Rerun this script a few times.

x1 = rnorm(100) x2 = rnorm(100) x3 = rnorm(100)

t = data.frame(a = (x1) , b = (x1+x2) , c = (x1+x2+x3)) plot(t) #sd(t) not working anymore sapply(t, sd)

```
source("Dataframe.R")
```



TODO Number 9

Add these lines to the script file of the previous section. Try to find out, either by experimenting or by using the help, what the meaning is of rgb, the last argument of rgb, lwd, pch, cex.

RGB = RGB Color Specification rgb(red, green, blue, alpha, names = NULL, maxColorValue = 1) maxColorValue: number giving the maximum of the color values range

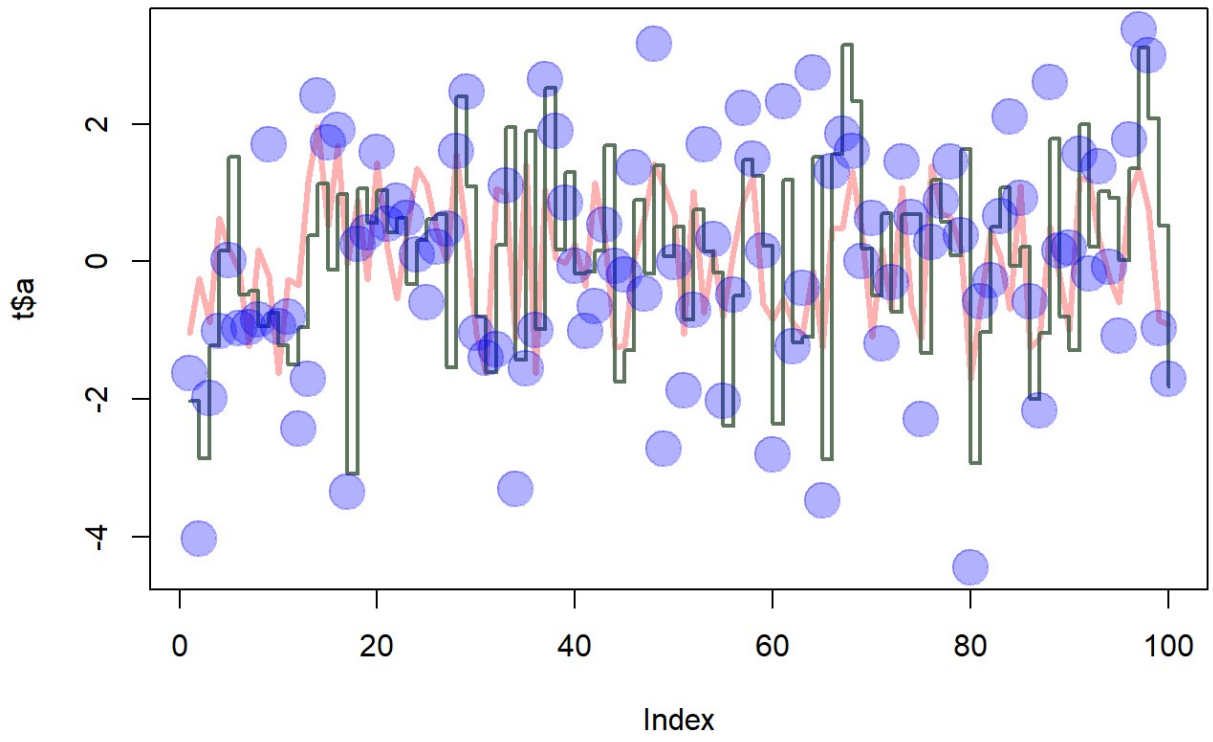
x1 = rnorm(100) x2 = rnorm(100) x3 = rnorm(100)

t = data.frame(a = (x1) , b = (x1+x2) , c = (x1+x2+x3))

plot(t) sapply(t, sd)

plot(t$a, type=" l ", ylim = range(t), lwd = 3, col = rgb(1, 0, 0, 0.3)) lines(t b, type="s", lwd=2, col=rgb(0.3,0.4,0.3,0.9)) points(t$c, pch=20, cex=4,col=rgb(0,0,1,0.3))

```
source("Dataframe2.R")
```



TODO Number 10

Make a file called tst1.txt in Notepad from the example in Figure 4 and store it in your working directory. Write a script to read it, to multiply the column called g by 5 and to store it as tst2.txt.

```
d2 = read.table(file="tst1.txt", header = TRUE)
write.table((d2$g * 5),file="tst2.txt",row.name=FALSE)
```

Contents of txt2

“x” 10 20 40 80 160 320

TODO Number 11

Compute the mean of the square root of a vector of 100 random numbers. What happens?

```
vec = rnorm(100)
mean(sqrt(vec))
```

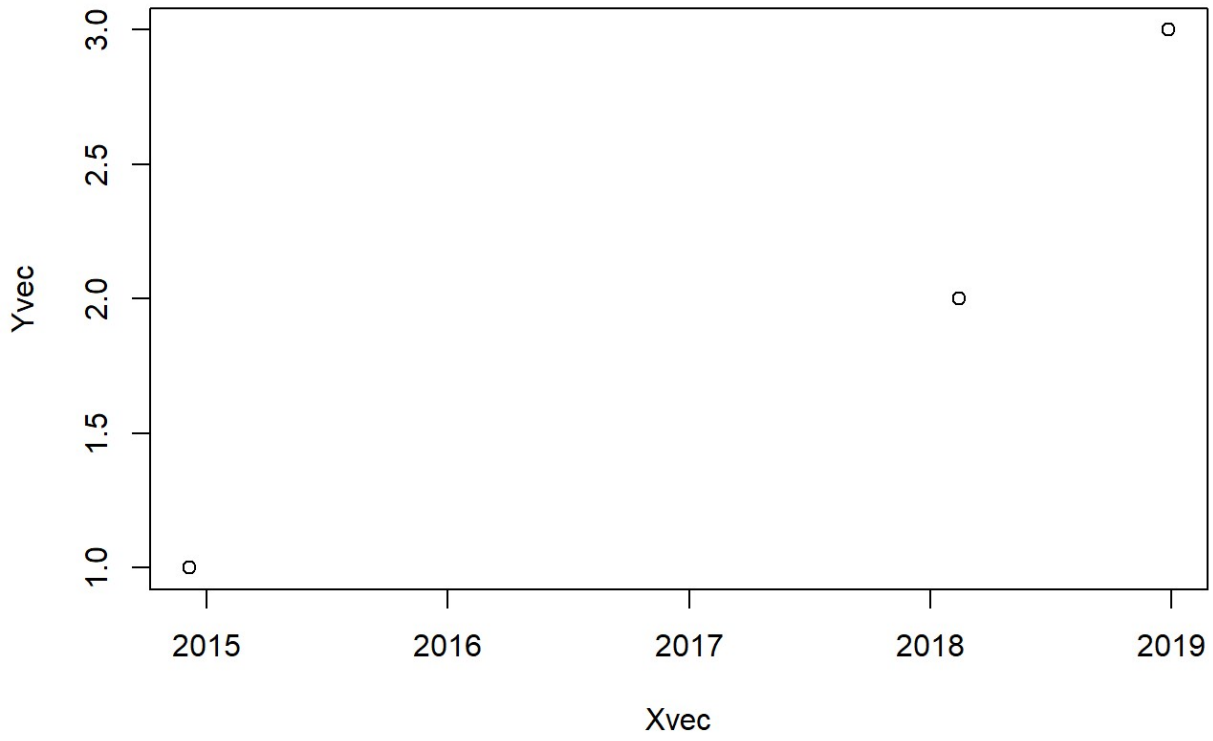
```
## Warning in sqrt(vec): NaNs produced
```

```
## [1] NaN
```

TODO Number 12

Make a graph with on the x-axis: today, Sinterklaas 2014 and your next birthday and on the y-axis the number of presents you expect on each of these days. Tip: make two vectors first.

```
Xvec = strptime(c("2018-02-14" , "2014-12-6", "2018-12-28"), "%Y-%m-%d")
Yvec = c(2,1,3)
plot(Xvec, Yvec)
```



TODO Number 13

Make a vector from 1 to 100. Make a for-loop which runs through the whole vector. Multiply the elements which are smaller than 5 and larger than 90 with 10 and the other elements with 0.1.

```
vec=seq(from=1, to=100, by=1)
s=c()
for(i in 1:100)
{
if(vec[i] < 5 | vec[i] > 90)
{
s[i]=vec[i]*10;
}

else
{
s[i]=vec[i]*0.1;
}
}
s
```

```
##   [1]   10.0   20.0   30.0   40.0    0.5    0.6    0.7    0.8    0.9    1.0
##  [11]    1.1    1.2    1.3    1.4    1.5    1.6    1.7    1.8    1.9    2.0
##  [21]    2.1    2.2    2.3    2.4    2.5    2.6    2.7    2.8    2.9    3.0
##  [31]    3.1    3.2    3.3    3.4    3.5    3.6    3.7    3.8    3.9    4.0
##  [41]    4.1    4.2    4.3    4.4    4.5    4.6    4.7    4.8    4.9    5.0
##  [51]    5.1    5.2    5.3    5.4    5.5    5.6    5.7    5.8    5.9    6.0
##  [61]    6.1    6.2    6.3    6.4    6.5    6.6    6.7    6.8    6.9    7.0
##  [71]    7.1    7.2    7.3    7.4    7.5    7.6    7.7    7.8    7.9    8.0
##  [81]    8.1    8.2    8.3    8.4    8.5    8.6    8.7    8.8    8.9    9.0
##  [91]  910.0  920.0  930.0  940.0  950.0  960.0  970.0  980.0  990.0 1000.0
```

TODO Number 14

Write a function for the previous ToDo, so that you can feed it any vector you like (as argument). Use a for-loop in the function to do the computation with each element. Use the standard R function length in the specification of the counter. a)

```
fun = function(arg1)
  {
  s=c()
  len = length(arg1)
  for(i in 1:len){
    if( arg1[i] < 5 | arg1[i] > 90)
      {
      s[i]=arg1[i]*10;
      }

    else
      {
        s[i]=arg1[i]*0.1;
        }
  }
  s
}

fun(c(1,2,3,4,5,6,7,8,9,10,91,92,93,94,95,96,97,98,99,100))
```

```
##  [1]   10.0   20.0   30.0   40.0    0.5    0.6    0.7    0.8    0.9    1.0
## [11]  910.0  920.0  930.0  940.0  950.0  960.0  970.0  980.0  990.0 1000.0
```

TODO FOOTNOTE extra

Actually, people often use more for-loops than necessary. The ToDo above can be done more easily and quickly without a for-loop but with regular vector computations.

```
fun = function(arg1)
  {

  first = arg1[1:4]
  middle = arg1[arg1 > 5 & arg1 < 90]
  last = arg1[arg1>90]

  s = c(first*10,middle*0.1,last*10)
  s
  }

fun(c(1,2,3,4,5,6,7,8,9,10,91,92,93,94,95,96,97,98,99,100))
```

```
##  [1]   10.0   20.0   30.0   40.0    0.6    0.7    0.8    0.9    1.0  910.0
## [11]  920.0  930.0  940.0  950.0  960.0  970.0  980.0  990.0 1000.0
```

Embedded code

The current time is Fri Feb 16 11:54:34 AM 2018.

The end